10783275

IDC-A3,AMD,M

simulating a set of API calls in accordance with the present invention is faster than executing the same API calls with fully implemented DLLs. Also, the virtual operating environment 106 of the present invention does not simulate all API calls supported in the related operating systems. API calls that are not indicative of malware and, as a result, are not considered "interesting" by the present invention, are not simulated.

~~Please amend the paragraph beginning on page 10, line 6, as follows:~~ IDC-A4,AMD

FIGURE [[7]] 7A is a flow diagram illustrative of a simulation routine 700 suitable for implementation by the computing device 100. At block 702, the simulation routine begins. As described above, the virtual operating environment 106 consists of software-generated components that simulate a specific operating system, such as the Win 32 operating system. The software-generated components include an interface that allows the virtual operating environment to be instantiated and receive and execute executables.

~~Please amend the paragraph beginning on page 12, line 12, as follows:~~ IDC-A5,AMD

If an API call requires a stub DLL for simulation, at block 718 (FIGURE 7B), the stack data structure 206 is queried for the reference information of the selected API. The reference information obtained from the stack data structure 206 permits identification of the correct stub DLL to load into the virtual address space 210.

Jm 3/12/08

~~Please amend the paragraph beginning on page 12, line~~ 16 ~~15, as follows:~~ IDC-A6,AMD,M

[[At]] As illustrated in FIGURE 7B, at block 720 an event is generated initiating the process of loading a stub DLL into the virtual address space 210. In some operating systems, such as the Win 32 operating systems, interactions between executables and computer hardware are coordinated by the operating system. For example, when an executable issues an API call requiring input, an event is generated and control of the hardware platform is transferred to the operating system. The operating system obtains data from the hardware platform and makes it available to the calling executable. FIGURES 3 and 4 and the accompanying text describe one

10783275

IDC-A6,AMD,M

example of when an operating system coordinates I/O after an event is generated with the loading of DLLs from a storage media 314 (i.e., input) into an executable's address space 318. Similarly the present invention generates an event when a stub DLL needs to be loaded to a location in memory available to the virtual operating environment 106, i.e., the virtual address space 210. FIGURES 5 and 6 and the accompanying text describe the process of loading a stub DLL from a storage media 314 into the virtual address space 210 after an event is generated.

~~Please amend the paragraph beginning on page 13, line 1, as follows:~~ IDC-A7,AMD

At decision block 722 depicted in FIGURE 7B, a test is conducted to determine whether the stub DLL that will simulate the selected API call is already loaded in the virtual address space 210. Since the virtual operating environment 106 simulates a sequence of API calls, the correct stub DLL may already be loaded into virtual address space 210. Stub DLLs that are already loaded in the virtual address space 210 are not loaded again.

~~Please amend the paragraph beginning on page 13, line 28, as follows:~~ IDC-A8,AMD

[[At]] Returning to FIGURE 7A, at decision block 728, a test is conducted to determine whether there are additional API calls that are potentially indicative of malware. As described above, API calls identified for execution are stored in a list. Contents of the list are sequentially traversed until all API calls have been executed in the virtual operating environment 106. If all API calls have been executed, at block 730 the output store is closed and at block 732 the routine terminates. If some API calls have not been executed, the routine cycles back to block 710, and blocks 710 through 728 are repeated until all required API calls have been executed.

~~Please amend the paragraph beginning on page 14, line 18, as follows:~~ 17 IDC-A9,AMD,M gm 3/12/08

The interface 200 of the virtual operating environment 106 allows virus scanning software to instantiate the virtual operating environment 106 and pass executables such as executable 108 to the virtual operating environment for execution. When executable 108 is